

⑫ 公開特許公報(A) 平4-186434

⑬ Int. Cl.⁵

G 06 F 9/44

識別記号

3 3 0 K

庁内整理番号

8724-5B

⑭ 公開 平成4年(1992)7月3日

審査請求 未請求 請求項の数 5 (全5頁)

⑮ 発明の名称 エキスパートシステム

⑯ 特 願 平2-313924

⑰ 出 願 平2(1990)11月21日

⑱ 発 明 者 福 田 光 子 茨城県日立市森山町1168番地 株式会社日立製作所エネルギー研究所内

⑲ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑳ 代 理 人 弁理士 小川 勝男 外2名

明 細 書

1. 発明の名称

エキスパートシステム

2. 特許請求の範囲

1. 与えられた問題に対する解を生成する推論手段と、推論に必要な知識を格納した知識ベースと、前記解に対する制約を格納した記憶装置を備えたエキスパートシステムにおいて、

各々の重要性に応じて制約に付与された数値から、生成される解による制約の充足度を求める手段を備え、完全に制約を充足する解が存在しない場合には、制約充足度を含んだ評価値により妥協案を求めて解とすることを特徴とするエキスパートシステム。

2. 請求項1において、

前記妥協案を求めるための評価値を、制約の充足度と解の探索効率の重み付きの和とするエキスパートシステム。

3. 請求項1において、前記制約充足度に下限を設け、前記充足度が下限値未満の場合は解とし

ないエキスパートシステム。

4. 請求項1において、前記評価値を前記制約充足度と探索効率の非線形関数で定義するシステム。

5. 請求項1において、前記評価値を算出する関数を複数用意して、システムの利用者が関数を選択する手段を備えたシステム。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、計画問題等、ある問題の解を生成する知識工学システムで、特に複数の制約を満たしながら問題解決を行う推論方式に関する。

〔従来の技術〕

従来のシステムでは、生成した解に対する制約は全て充足されなければならない、現実の問題としても軽微な影響しか及ぼさない程度の制約に違反しても解は生成されない。これに対し、特開昭64-21537号公報では制約に優先順位を設けて、完全に制約を充足する解が存在しない場合、優先順位の低い順に制約を取り除いて、優先順位の高い制

約だけ充足する解を、妥協案として提示するシステムが開示されている。(特開昭64-21537号公報)

〔発明が解決しようとする課題〕

上記従来技術は制約に優先順位を付与することによって、重要性の低い制約によって解が生成できなくなること回避したが、ここで指定されるのは順序のみであって、各制約の相対的な重要性は反映していない。この方法では優先順位の低い制約から、順次、放棄して、完全に充足できる制約の部分集合を得るので、ある一つの制約の違反により、それ以下の順位の制約が全て放棄されてしまう。こうして生成される解は必要以上に制約を取除いて得られた解である可能性があり、特に、制約の重要性が僅差である場合は生成される解が真に適切な妥協案でない可能性が大きい。

本発明の目的は、不必要に制約を取り除くことなく、常に、適切な妥協案を提示できる方法を提供することにある。

〔課題を解決するための手段〕

る。知識ベース1には与えられる問題を解決するのに必要な知識が、例えば、if-then ルールの形で格納されており、この知識から推論手段5が解を生成する。生成される解に対する制約2aは制約記憶装置2に、その重要度2bと一組にして格納されている。推論過程の履歴は管理手段4を通して記憶装置3に保存され、推論失敗、又は、制約違反によるバックトラック時に利用される。履歴には充足度算出手段8によって算出された制約の充足度を反映した評価値が、制約違反の起った状態のそれぞれに付加されている。この評価値は、各制約の充足度と、制約に違反した時点でどの程度まで問題が解けていたかを示す指標値から算出する。問題とその解妥協案選定のパラメータ等は入出力装置6、7a、7bを介して行う。

本システムの動作を第2図、第3図により説明する。第2図は処理概要のフローチャート、第3図は推論の過程を表した説明図である。この実施例のシステムでは、初期状態と目標状態を与えられて、初期状態を目標状態に移す作業計画を推論

上記目的を達成するために、本考案は各制約にその重要性に応じた値を重要度として付与することにより各制約間の相対的な重要性を反映できるようにし、また、必要以上に制約を取り除くことなく解を求めるために、各制約の重要度から制約全体の充足度を算出する手段を設けて、より良く制約を充足する解を選ぶ評価値を生成する。

〔作用〕

各制約の重要度に基づいた充足度算出手段は、最終的な解を得るに至るまでの過程で、中間的な解を得る度に、中間解による制約の充足度を算出する。別の知識を適用した推論で得られる、別の中間解が存在すれば、各中間解について、解による制約の充足度と最終的な解と中間解の近さの指標から成る評価値を求め、最良の評価値を与える中間解を選んで推論していくので、計算コストに見合った範囲で最良の妥協案が得られる。

〔実施例〕

以下、本考案の一実施例を説明する。

第1図は本考案によるシステムの全体構成であ

する。各作業は状態を変更する作用素として表され、推論手段5は適当な作業素を推論することで作業計画を生成する。

このシステムは、まず、問題を入力されると初期状態 S_0 を設定し(処理21)、推論過程の履歴を表す木の根30として記憶する。次に S_0 に作用させる適当な作用素31を推論して、次の状態 S_1 を求める(処理22)。推論が成功すれば(処理23)、求まった S_1 は履歴を表す木のノード32として追加される。次に、 S_1 で各制約が充足されているか判定して(処理24)、充足していれば更に S_1 が目標状態を達成したか判定する(処理25)。ここではまだ目標状態を達成していないので、推論を続行し、次の状態33(S_2)を得る。ここで S_2 がある制約 C_i に違反すると、 C_i を除いた制約を全て充足できると仮定して推論を続行した場合の制約充足度 X_i を求める(処理26)。充足度 X_i は、制約 C_i の重要度 W_i から、例えば、 $A = \{j \mid C_j \text{ は任意の制約} \}$ 、 $B_i = \{n \mid C_n \text{ は状態 } S_2 \text{ で充足できない任}$

意の制約)として、 $X_i = 1 - \sum_{k \in K_i} W_k / \sum_{k \in K} W_k$ とする。

更に、 S_i がどの程度目標状態に近いかを表す指標 $P(S_i)$ を求めて、 S_i の評価値 $V(S_i) = t_i X_i + t_p P(S_i)$ を算出する。ここで定数 t_i 、 t_p は妥協案とする解の探索に、制約の充足度と目標到達までに費す計算量をどの位の比率で重視して妥協案を求めるかのトレードオフを表す値である。求めた評価値 $V(S_i)$ と、 S_i が違反する制約は S_i のノードの属性として履歴に登録する(処理 27)。このノードは、制約を完全に充足する解が存在しない場合に、制約を取り除いて妥協案を求めるための妥協点になる。

次に、別解がないか、 S_i から別のルールを適用して 34 (S_j) を求める。ここで S_i もまた制約 C_i に違反すると、 S_i と同様に評価値を求めて履歴に登録する。

更に S_i に戻って別解を求め、適用できるルールが尽きれば、それまでに求めた妥協点 S_i 、 S_j から評価値の大きい方を選択する(処理 28)。 S_i 、 S_j は目標状態との差の指標は同じだが、違

反した制約の重要度の差によって S_i が選択されたとする。 S_i を選択したことで、妥協案の充足すべき制約は初期の制約から C_i を除いたものになる(処理 29)。妥協案を求める推論は、再び、同様の処理をくり返して次の状態 35 (S_k) を得る。これが別の制約 C_k に違反すると、充足度 X_k 、指標値 $P(S_k)$ から評価値 $V(S_k)$ が算出される。ここで $X_k > X_i$ 、 $P(S_k) \leq P(S_i)$ である。 S_i から推論できる状態が他になければ、新しく推論を続行する妥協点に S_k と先に棄却された S_i のうちの評価値の高い方を選択する。ここでは制約の充足を重視する例として、 S_i を棄却し、 S_k を選択する。

33 (S_i) を選択することにより、それまで除かれていた制約 C_i を再び加え、 S_i で充足できない制約 C_i を取り除く(処理 29)。 S_i から推論を続行し、新たに制約に違反する状態 36 が発生すれば、まず別のルールを適用して、制約に違反せずに推論を続行できる状態を求め、再び、制約が充足できない状態に到れば新たな評価値を算出

し、先に棄却された妥協点となる状態 35、36 と比較して評価値の高い状態に戻って推論を続行し、目標状態に到る。

こうして妥協案として、31、38~41 の作業系列で表された作業計画を得る。これは状態 33 (S_i) で違反した制約 C_i を取り除いた場合の解である。特開昭 64-21537 号公報のように、優先順位による方法では、状態 S_i 、 S_j で違反した制約 C_i 、 C_j が比較的上位の優先度を与えられていると、それ以下の制約も全て取り除かれてしまい、得られた妥協案がどの程度有効な解であるか非常に信頼性が低い。また、優先度の低い順に取り除くため、この例題の S_i 、 S_j 、 S_k が違反した制約 C_i 、 C_j 、 C_k の重要性が $C_i < C_j < C_k$ 、かつ、 $C_i < C_j + C_k$ であると、優先順位は C_i の方が高いために、制約 C_i 、 C_j を取り除いた場合の解を求めるので、妥協案が充足できる制約はより限られたものになってしまう。本システムでは、常に、制約全体を考慮した充足度を用いるので、このような失敗がない。

また、妥協案を求めるのに要する計算量の予測量が、目標状態にどれだけ近付いたかという指標値によって、妥協案選択の評価値に反映されているので、制約の充足度と計算量について適当なトレードオフ値を設定することにより、制約の充足を重視する場合には、計算量がある程度増えても充足度の大きい妥協案を推論し、計算量を重視する場合はある程度充足度が低くても少い計算量で得られる妥協案を提示するというように、常に使用者の望むレベルで計算量に見合った妥協案が得られる。

第二の実施例として、制約充足度に下限を設けて、下限未満の充足度しか得られない場合は解としない方法もある。この実施例の処理は、第一の実施例とほぼ同じだが、問題の入力と共に任意の下限値を入力し、制約に違反した場合に充足度が下限値未満であれば評価値は算出せず、履歴には推論続行のできない状態として登録する。これにより、得られる解が最低限充足する制約が保障される。

第三の実施例として、第一の実施例で充足度 X_i 、解の探索効率の指標値 P_i 、ある定数 t_1 、 t_2 をとって $V_i = t_1 X_i + t_2 P_i$ とした評価値 V_i を非線形関数で定義する。例えば、 $V_i = t_1 \log(X_i - a) + t_2 \exp(P_i - b)$ として適当な定数 a 、 b を選べば、探索初期の $P_i \leq b$ の時点では充足度の差に敏感な評価値が、探索が進んで、かつ、ある程度の充足度が確保されていれば探索効率に敏感な評価値が得られる。このような評価値を用いれば比較的充足度の高い解が効率良く求められる。

更に、第四の実施例としては、第一、第三の実施例で示したような評価関数を複数用意し、システムの利用者が任意に評価関数を選択する手段を設ける。このようにすれば、問題や制約の性質に合わせた評価値が得られるので、より有効な解を求めることができる。

〔発明の効果〕

本発明によれば、完全には制約を充足できない問題に対しても、一部の制約を取り除いて求まる解から妥協案が得られ、しかも、妥協案の選択は、

各制約の重要性を反映した制約全体の充足度に基づくので、可能な限り制約を充足する良い妥協案が得られる。

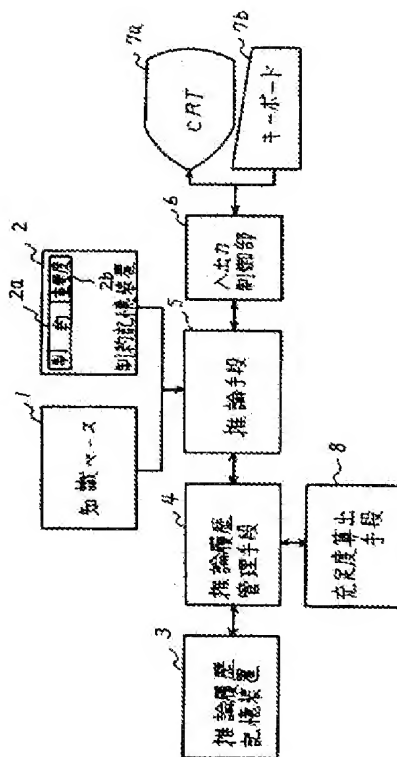
4. 図面の簡単な説明

第1図は本発明の一実施例のブロック図、第2図は第1図のシステムの処理のフローチャート、第3図は例題についての推論の過程を示した説明図である。

1…推論に用いる知識ベース、2a…推論して得られる解に対する制約、2b…当該制約の重要度、3…推論履歴の記憶装置、4…推論履歴の管理手段、5…推論手段、6…各制約の重要度に基づく解の制約充足度算出手段。

代理人 弁理士 小川勝男

第1図



第2図

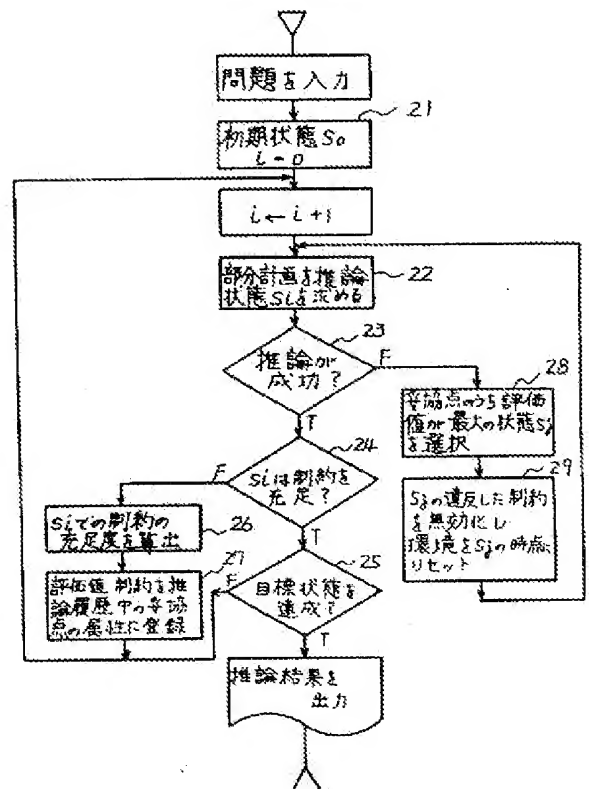


図 3

